**Department of Electrical Engineering**

|  |  |
| --- | --- |
| **Faculty Member: Sir Haasaan Khaliq** | **Dated: 2/24/2023** |
| **Semester:6th** | **Section: C** |

EE-357 Computer and Communication Networks

Experiment - 3

**General introduction to Wireshark and networking**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **PLO5/**  **CLO4** | **PLO9/**  **CLO5** |
| **Name** | **Reg. No** | **Modern Tool Usage** | **Individual and Team Work** |
|  |  | **10 Marks** | **5 Marks** |
| **Muhammad Ahmed Mohsin** | **333060** |  |  |
| **Amina Bashir** | **343489** |  |  |
| **Imran Haider** | **332569** |  |  |
|  |  |  |  |

Table of Contents

[2 Objective of this lab: 3](#_Toc128151497)

[3 Instructions: 3](#_Toc128151498)

[4 Background: 3](#_Toc128151499)

[4.1 Introduction to Networking: 4](#_Toc128151500)

[4.2 What is a protocol analyzer? 4](#_Toc128151501)

[4.3 Introduction to Wireshark 4](#_Toc128151502)

[5 Steps for performing this lab: 8](#_Toc128151503)

[5.1 OUTPUT 10](#_Toc128151504)

[5.2 OUPUT 11](#_Toc128151505)

[5.3 OUTPUT 12](#_Toc128151506)

[6 Questions: 13](#_Toc128151507)

[6.1 ANSWER 14](#_Toc128151508)

[6.2 Answer: 15](#_Toc128151509)

[7 Additional Lab Task 16](#_Toc128151510)

[8 Conclusion: 17](#_Toc128151511)

**EXPERIMENT NO 5**

**General introduction to Wireshark and networking**

# Objective of this lab:

The basic purpose of this lab is to introduce you to Wireshark, a popular protocol analyzer. By the end of this lab you will be familiar to its environment and will know how to capture and interactively browse the traffic running on a computer network using it.

# Instructions:

1. Read carefully before starting the lab.
2. These exercises are to be done individually.
3. You are supposed to provide the answers to the questions listed at the end of this document and upload this completed document to your course’s LMS site.
4. Avoid plagiarism by copying from the Internet or from your peers. You may refer to source/ text but you must paraphrase the original work.

# Background:

A protocol analyzer is a tool that can be used to inspect what exactly is happening on a network with respect to traffic flow. For example, if your TCP/IP sessions are "hanging", a protocol analyzer can show which system sent the last packet, and which system failed to respond. If you are experiencing slow screen updates, a protocol analyzer can display delta time stamps and show which system is waiting for packets, and which system is slow to respond.

A protocol analyzer can show runaway traffic (broadcast or multicast storms) and its origin, system errors and retries, and whether a station is sending, trying to send, or only seeming to communicate. You will get information that is otherwise unavailable, which results in more efficient troubleshooting and better LAN health.

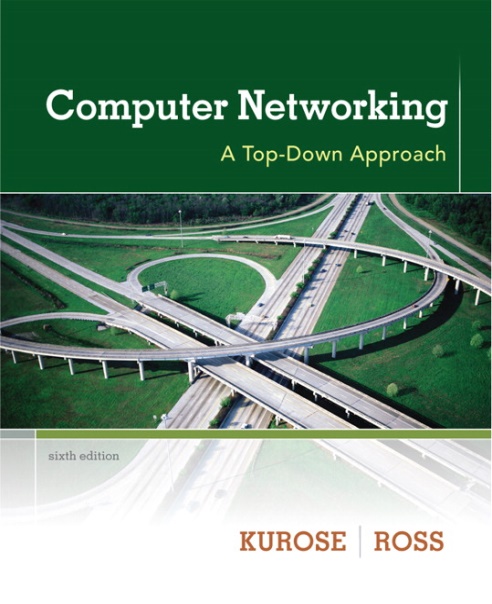
## Introduction to Networking:

A **computer network**, often simply referred to as a network, is a collection of hardware components and computers interconnected by communication channels that allow sharing of resources and information. In the world of computers, networking is the practice of linking two or more computing devices together for the purpose of sharing data. In networking, the communication language used by computer devices is called the [protocol](http://compnetworking.about.com/library/glossary/bldef-protocol.htm). Yet another way to classify computer networks is by the set of protocols they support. Networks often implement multiple protocols to support specific applications.

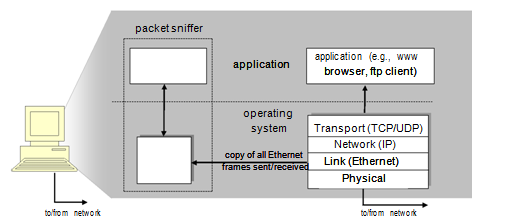
## What is a protocol analyzer?

Protocol analyzers capture conversations between two or more systems or devices. A protocol analyzer not only captures the traffic, it also decodes (interprets) the traffic. Decoding allows you to view the conversation in English, as opposed to binary language. A sophisticated protocol analyzer will also provide statistics and trend information on the captured traffic. Protocol analyzers provide information about the traffic flow on your local area network (LAN), from which you can view device-specific information.

## Introduction to Wireshark

**Wireshark** is a [free and open-source](http://en.wikipedia.org/wiki/Free_and_open_source_software) [packet analyzer](http://en.wikipedia.org/wiki/Packet_analyzer), used for [network](http://en.wikipedia.org/wiki/Computer_network) troubleshooting, analysis, software and [communications protocol](http://en.wikipedia.org/wiki/Communications_protocol) development, and education.

The basic tool for observing the messages exchanged between executing protocol entities is called a **packet sniffer**. As the name suggests, a packet sniffer captures (“sniffs”) messages being sent/received from/by your computer; it will also typically store and/or display the contents of the various protocol fields in these captured messages. A packet sniffer itself is *passive*. It observes messages being sent and received by applications and protocols running on your computer, but never sends packets itself. Similarly, received packets are never explicitly addressed to the packet sniffer. Instead, a packet sniffer receives a copy of packets that are sent/ received from/by application and protocols executing on your machine.

Figure 1 shows the structure of a packet sniffer. At the right of Figure 1 are the protocols (in this case, Internet protocols) and applications (such as a web browser or ftp client) that normally run on your computer. The packet sniffer, shown within the dashed rectangle in Figure 1 is an addition to the usual software in your computer, and consists of two parts. The **packet capture library** receives a copy of every link-layer frame that is sent from or receivedby your computer. Messages exchanged by higher layer protocols such as HTTP, FTP, TCP, UDP, DNS, or IP all are eventually encapsulated in link-layer frames that are transmitted over physical media such as an Ethernet cable. In Figure 1, the assumed physical media is an Ethernet, and so all upper-layer protocols are eventually encapsulated within an Ethernet frame. Capturing all link-layer frames thus gives you all messages sent/received from/by all protocols and applications executing in your computer.

**Figure 1**: *Packet sniffer structure*

The second component of a packet sniffer is the **packet analyzer**, which displays the contents of all fields within a protocol message. In order to do so, the packet analyzer must “understand” the structure of all messages exchanged by protocols. For example, suppose we are interested in displaying the various fields in messages exchanged by the HTTP protocol in Figure 1. The packet analyzer understands the format of Ethernet frames, and so can identify the IP datagram within an Ethernet frame. It also understands the IP datagram format, so that it can extract the TCP segment within the IP datagram. Finally, it understands the TCP segment structure, so it can extract the HTTP message contained in the TCP segment. Finally, it understands the HTTP protocol and so, for example, knows that the first bytes of an HTTP message will contain the string “GET,” “POST,” or “HEAD,”.

We will be using the Wireshark packet sniffer [http://www.wireshark.org/] for these labs, allowing us to display the contents of messages being sent/ received from/by protocols at different levels of the protocol stack. (Technically speaking, Wireshark is a packet analyzer that uses a packet capture library in your computer). Wireshark is a free network protocol analyzer that runs on Windows, Linux/Unix, and Mac computers. It’s an ideal packet analyzer for our labs – it is stable, has a large user base and well-documented support that includes a comprehensive user-guide (<http://www.wireshark.org/docs/wsug_html_chunked/>), man pages (<http://www.wireshark.org/docs/man-pages/>), and a FAQ (<http://www.wireshark.org/faq.html>), rich functionality that includes the capability to analyze more than 500 protocols, and a well-designed user interface. It operates in computers using Ethernet, Token-Ring, FDDI, serial (PPP and SLIP), 802.11 wireless LANs and ATM connections (if the OS on which it’s running allows Wireshark to do so).

**3.1 Getting Wireshark**

In order to run Wireshark, you will need to have access to a computer that supports both Wireshark and the *libpcap* or *WinPCap* packet capture library. The *libpcap* software will be installed for you alongside Wireshark automatically. See http://www.wireshark.org/download.html for a list of supported operating systems and download sites

Download and install the Wireshark software:

* Go to http://www.wireshark.org/download.html and download and install the Wireshark binary for your computer. Wireshark can be installed on both Windows and Linux. See the documentation page of Wireshark for more details.
* Download the Wireshark user guide.

The Wireshark FAQ has a number of helpful hints and interesting tidbits of information, particularly if you have trouble installing or running Wireshark.

* 1. **Running Wireshark**

On *Windows*, you should be able be able to find the link by clicking on the Start option of the Windows taskbar and thereby finding the wireshark program in All Programs.

*On Linux machines*, wireshark can be run by typing “wireshark” at the command prompt (in case there is a problem with your path, type “*which wireshark”* that would show path /usr/bin/wireshark where wireshark is typically installed). When you run the Wireshark program, the Wireshark graphical user interface shown in Figure 2 will be displayed. Initially, no data will be displayed in the various windows.

The Wireshark interface has five major components:

* The **command menus** are standard pull down menus located at the top of the window. Of interest to us is the File and Capture menus. The File menu allows you to save captured packet data or open a file containing previously captured packet data, and exits the Wireshark application. The Capture menu allows you to begin packet capture.
* **The packet-listing window** displays a one-line summary for each packet captured,including the packet number (assigned by Wireshark; this is not a packet number contained in any protocol’s header), the time at which the packet was captured, the packet’s source and destination addresses, the protocol type, and protocol-specific information contained in the packet. The packet listing can be sorted according to any of these categories by clicking on a column name. The protocol type field lists the highest-level protocol that sent or received this packet, i.e., the protocol that is the source or ultimate sink for this packet.
* The **packet-header details window** provides details about the packet selected (highlighted) in the packet-listing window. (To select a packet in the packet-listing window, place the cursor over the packet’s one-line summary in the packet-listing window and click with the left mouse button.). These details include information about the Ethernet frame and IP datagram that contains this packet. The amount of Ethernet and IP-layer detail displayed can be expanded or minimized by clicking on the right-pointing or down-pointing arrowhead to the left of the Ethernet frame or IP datagram line in the packet details window. If the packet has been carried over TCP or UDP, TCP or UDP details will also be displayed, which can similarly be expanded or minimized. Finally, details about the highest-level protocol that sent or received this packet are also provided.
* The **packet-contents window** displays the entire contents of the captured frame, in both ASCII and hexadecimal format.
* Towards the top of the Wireshark graphical user interface, is the **packet display filter** **field**, into which a protocol name or other information can be entered in order to filterthe information displayed in the packet-listing window (and hence the packet-header and packet-contents windows). In the example below, we’ll use the packet-display filter field to have Wireshark hide (not display) packets except those that correspond to HTTP messages.

# Steps for performing this lab:

The best way to learn about any new piece of software is to try it out! We’ll assume that your computer is connected to the Internet via a wired Ethernet interface. Do the following:

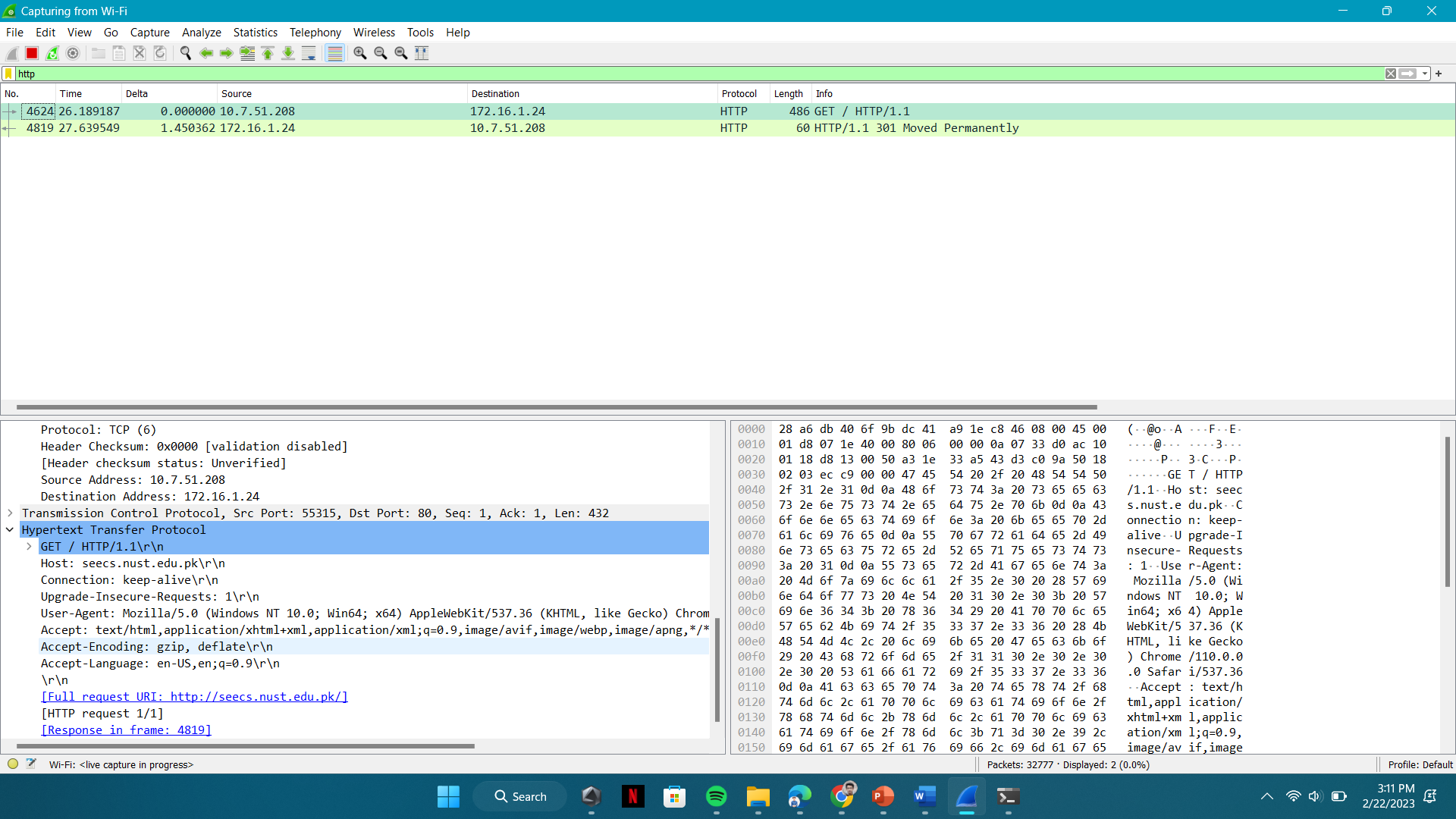
1. ***Start up your favorite web browser***, which will display your selected homepage.
2. ***Start up the Wireshark software.*** You will initially see a window similar to that shown in Figure 2, except that no packet data will be displayed in the packet-listing, packet-header, or packet-contents window, since Wireshark has not yet begun capturing packets.
3. ***To begin packet capture,*** select the Capture pull down menu and select *Options.* This will cause the “Wireshark: Capture Options” window to be displayed, as shown in Figure 3.
4. ***Selecting the network interface on which packets would be captured:*** You can use most of the default values in this window, but uncheck “Hide capture info dialog” under Display Options. The network interfaces (i.e., the physical connections) that your computer has to the network will be shown in the Interface pull down menu at the top of the Capture Options window. In case your computer has more than one active network interface (e.g., if you have both a wireless and a wired Ethernet connection), you will need to select an interface that is being used to send and receive packets (most likely the wired interface). After selecting the network interface (or using the default interface chosen by Wireshark), click Start. Packet capture will now begin - Wireshark is now capturing all packets being sent/received from/ by your computer!
5. Once you begin packet capture, a packet capture summary window will appear, as shown in Figure 4. This window summarizes the number of packets of various types that are being captured, and (importantly!) contains the *Stop* button that will allow you to stop packet capture. Don’t stop packet capture yet.

***6. Capturing an HTTP interaction on Wireshark:*** While Wireshark is running, enter the URL: <http://seecs.nust.edu.pk/> and have that page displayed in your browser. In order to display this page, your browser will contact the HTTP server at <http://seecs.edu.pk>, and exchange HTTP messages with the server in order to download this page, as discussed. Wireshark will capture the Ethernet frames containing these HTTP messages.

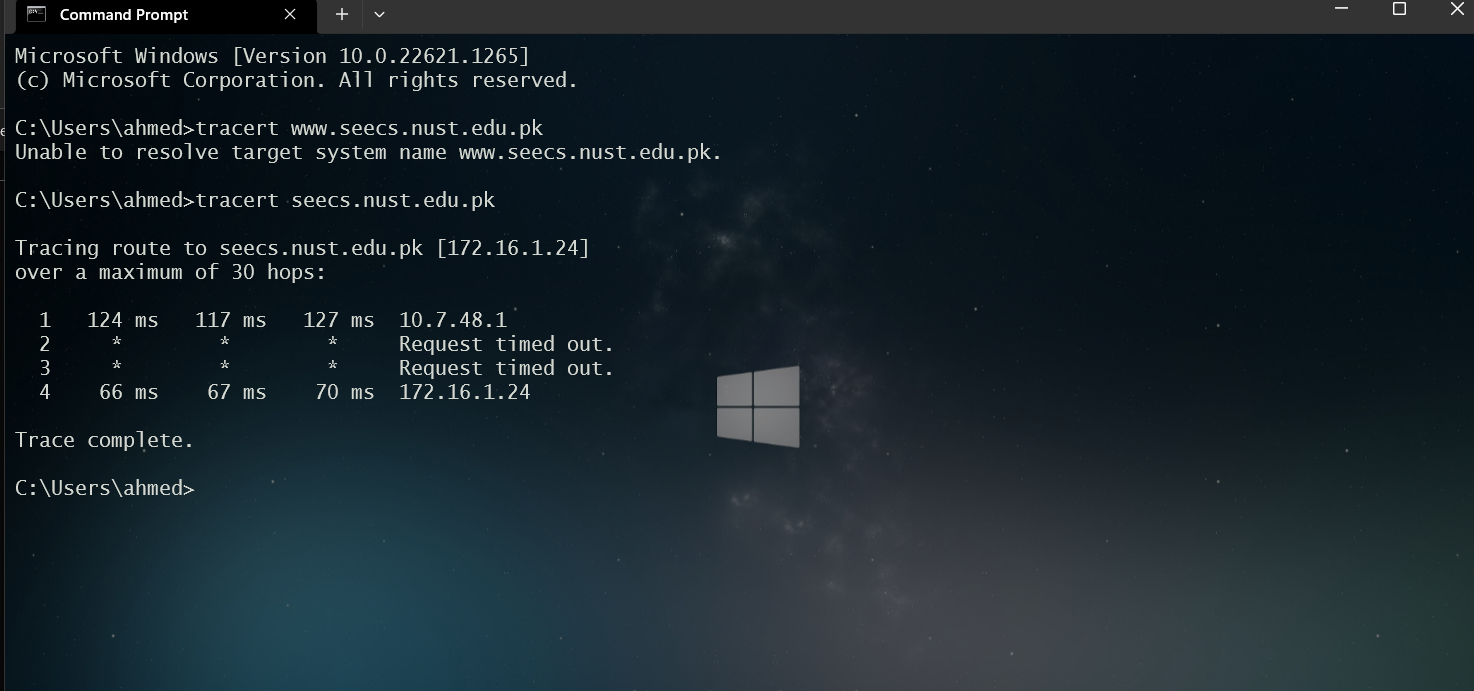
1. ***Stopping the capture and inspecting captured packets:*** After your browser has displayed the page, stop Wireshark packet capture by selecting stop in the Wireshark capture window. This will cause the Wireshark capture window to disappear and the main Wireshark window to display all packets captured since you began packet capture. The main Wireshark window should now look similar to Figure 2. You now have live packet data that contains all protocol messages exchanged between your computer and other network entities! The HTTP message exchanges with the seecs.nust.edu.pk web server should appear somewhere in the listing of packets captured. But there will be many other types of packets displayed as well (see, e.g., the many different protocol types shown in the *Protocol* column in Figure 2). Even though the only action you took was to download a web page, there were evidently many other protocols running on your computer that are unseen by the user. We’ll learn much more about these protocols as we progress through the text! For now, you should just be aware that there is often much more going on than “meets the eye”.
2. ***Filtering:*** Type in “http” (without the quotes, and in lower case – all protocol names are in lower case in Wireshark) into the display filter specification window at the top of the main Wireshark window. Then select *Apply* (to the right of where you entered “http”). This will cause only HTTP message to be displayed in the packet-listing window.

## OUTPUT

The packet traced by wire shark is as shown:

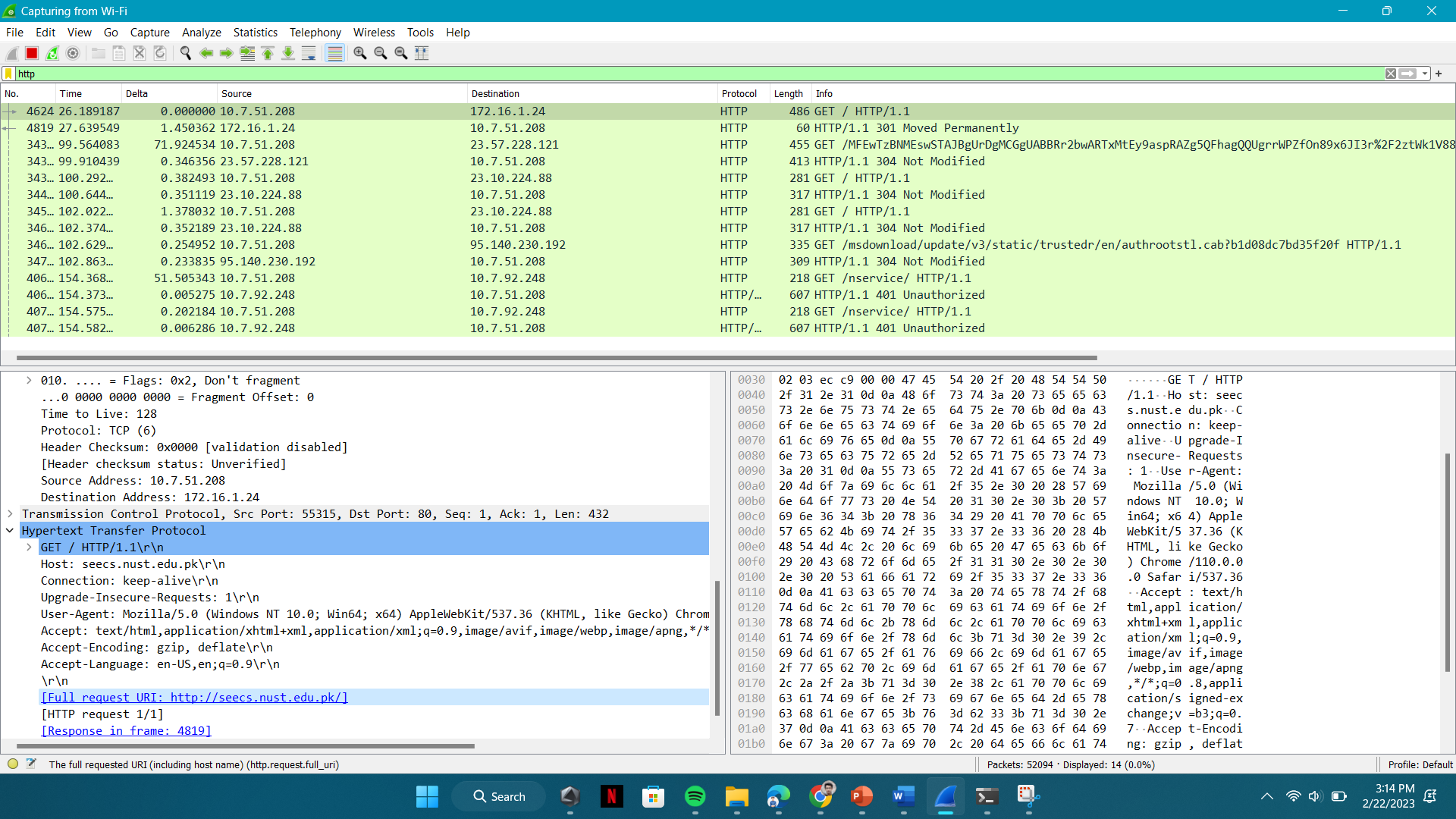


This is the same IP address of seecs.nust.edu.pk as observed by tracert command:



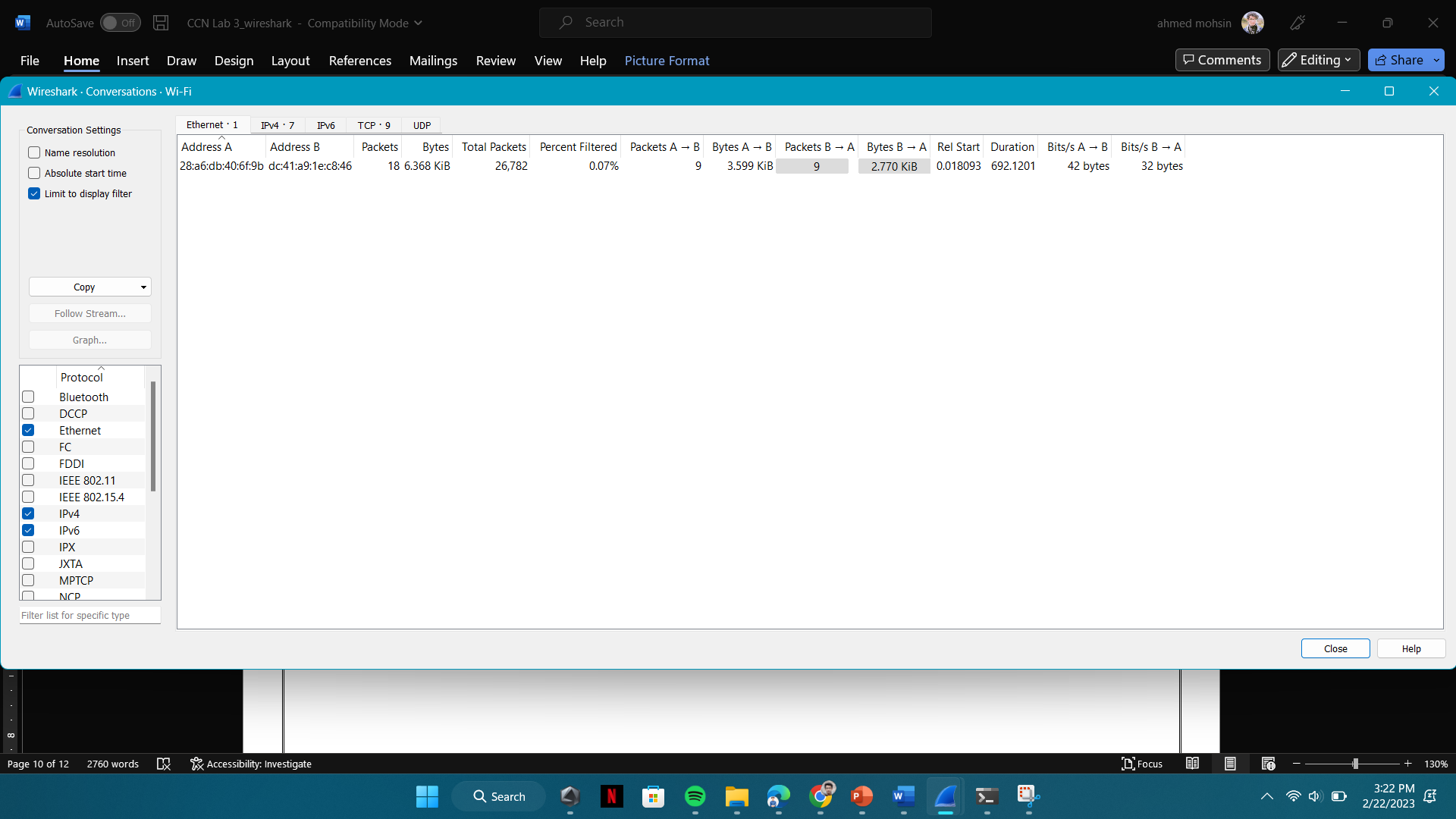
1. ***Details of a packet:*** Select the first http message shown in the packet-listing window. This should be the HTTP GET message that was sent from your computer to the seecs.nust.edu.pk HTTP server. When you select the HTTP GET message, the Ethernet frame, IP datagram, TCP segment, and HTTP message header information will be displayed in the packet-header window. By clicking on right-pointing and down-pointing arrows heads to the left side of the packet details window, *minimize* the amount of Frame, Ethernet, Internet Protocol, and Transmission Control Protocol information displayed. *Maximize* the amount information displayed about the HTTP protocol. Your Wireshark display should now look roughly as shown in Figure 5. (Note, in particular, the minimized amount of protocol information for all protocols except HTTP, and the maximized amount of protocol information for HTTP in the packet-header window).

## OUPUT

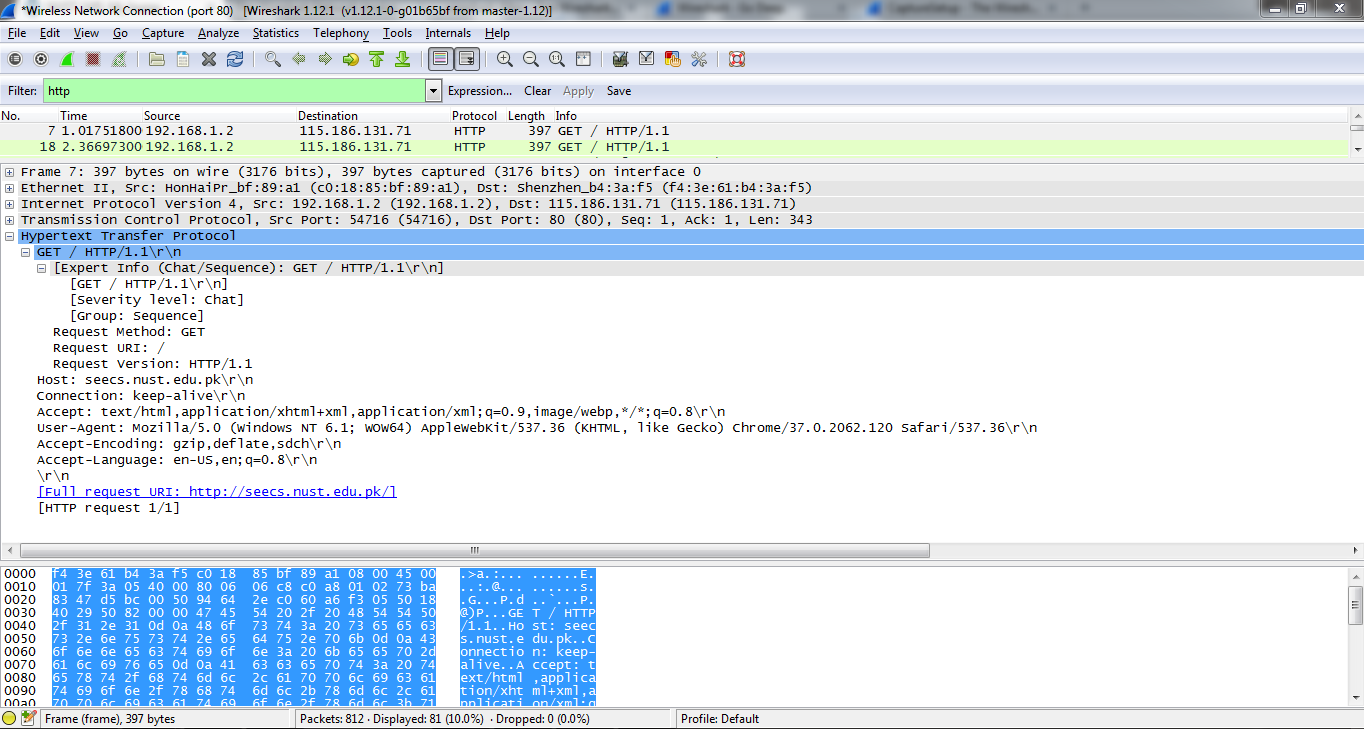


1. ***Statistics of packet captured:*** Click on the ‘Statistics’ option on the upper toolbar of Wireshark to explore the various ways in which statistics may be obtained about network traffic. Explore specifically the ‘Conversation’ options in ‘Statistics’ option on the upper toolbar of Wireshark.

## OUTPUT



1. ***Obtaining credit for this lab:*** Answer the questions listed at the end of this lab. Please note that this is an individual activity and every student must upload the answer file (after duly filling in the answers) through the appropriate link at your LMS course site for the specific date of your lab (an upload link would be made available) to obtain credit. Please clarify with your instructor/ lab engineer if you have any queries.

**Figure 5:** *Wireshark display after step 9*

# Questions:

1. ***Finding IP address of your machine in Wireshark:*** What is the IP address of ‘alibaba.com’? What is the IP address of your computer? How did you find it in Wireshark? Compare the IP address of your machine by using ipconfig command.

## ANSWER

**To find the IP address of 'alibaba.com':**

Open the command prompt or terminal on your computer. Type 'nslookup alibaba.com' and press Enter. The IP address associated with the domain name 'alibaba.com' will be displayed.

**To find the IP address of your computer:**

Open the command prompt or terminal on your computer. Type 'ipconfig' and press Enter. The IP address of your computer will be displayed under the 'IPv4 Address' or 'IP Address' field.

The IP address of my PC using wire shark is:

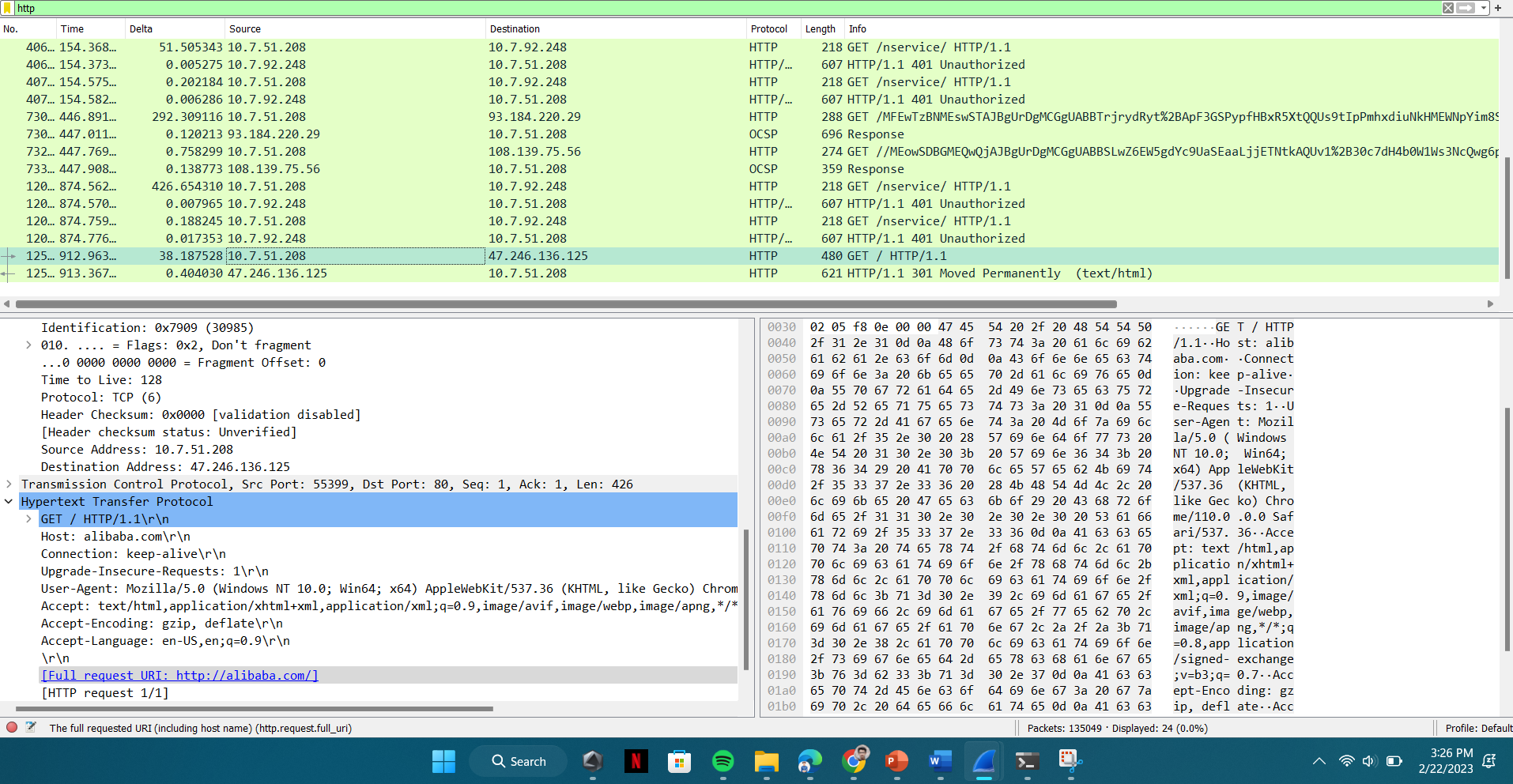


The IP address of my pc using tracert command is:

Text

Description automatically generated

The first hop shows the source PC and for this case is our own PC;s IP address.



1. What is the ***port number*** used by the HTTP server ‘alibaba.com’. How did you note it in Wireshark?

## Answer:

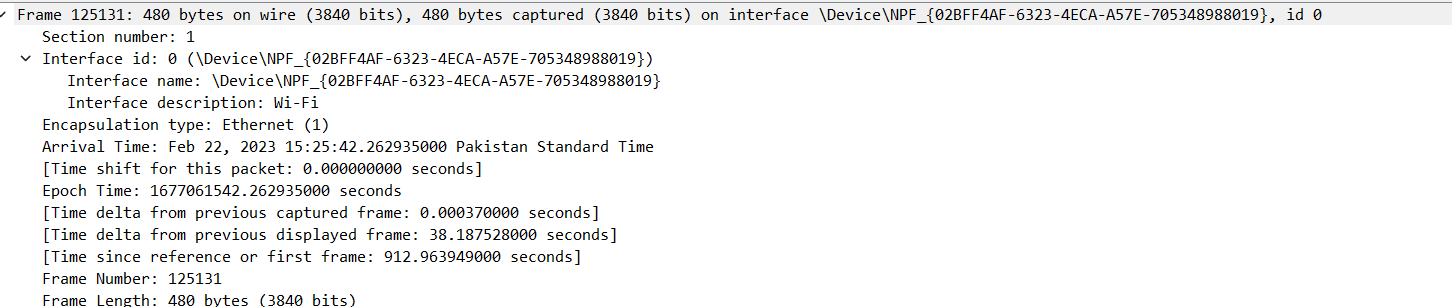
We can see the port number under the transmission control protocol as shown in the screenshot below:

Graphical user interface, text, application

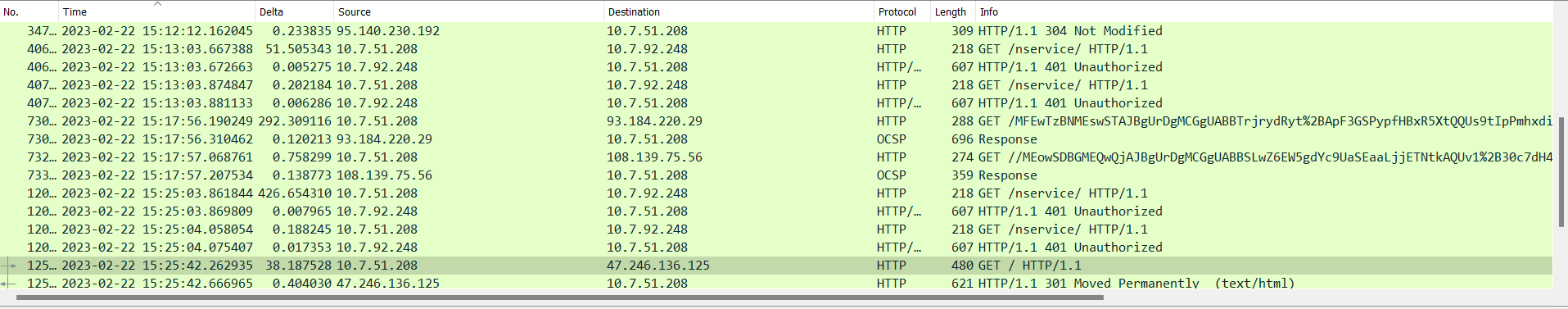
Description automatically generated

1. ***Delay between request and reply:*** How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet-listing window is the amount of time, in seconds, since Wireshark tracing began. To display the Time field in time-of-day format, select the Wireshark *View* pull down menu, then select Time *Display Format*, then select *Time-of-day*.)

The time taken by the packet to be send and received is sown below in the screen shot as “time of day” format.



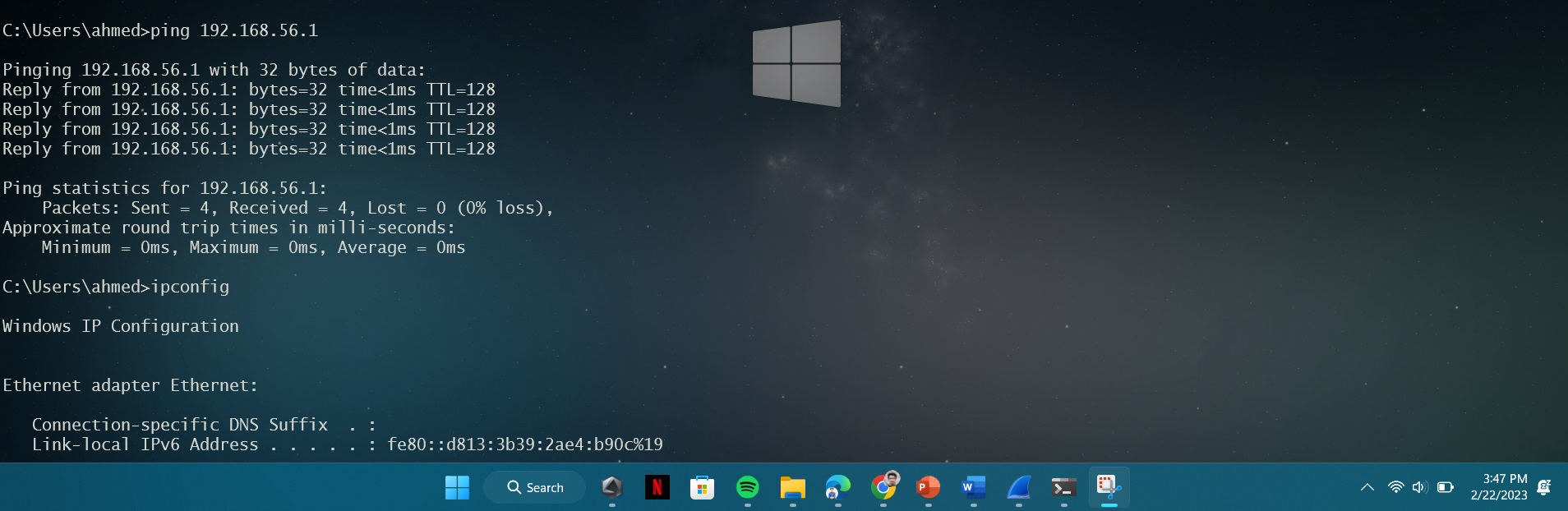
The time it took as displayed by wire shark time column is as:



# Additional Lab Task

**Pinging Imran Haider PC:**

I Pinged Imran Haider’s laptop using my PC by using the ping command and it worked.



# Conclusion:

In conclusion, this lab provided us with valuable insights into the powerful network analysis tool **Wireshark**. Through this lab, we were able to learn how to apply filters to analyze network traffic, identify the packets sent and received during the communication process, and extract valuable information from those packets. By tracing the packets of the site alibaba.com, we gained a better understanding of how internet protocols work, and how information is transmitted and received over the network. We were able to observe the flow of data packets, their structure, and the information they carry. Moreover, by answering theoretical questions related to network protocols, we were able to enhance our understanding of the underlying mechanisms and principles that govern network communication.